

Kukakuka: An Online Environment for Artifact-Centered Discourse

Daniel Suthers and Jun Xu

Laboratory for Interactive Learning Technologies
Department of Information and Computer Sciences
University of Hawai'i at Manoa
1680 East West Road, POST 309
Honolulu, HI 96822, USA
1-808-956-3890
suthers@hawaii.edu

Abstract

Most human discourse in learning and work relies on external artifacts as resources or aids for conversation, but many online learning environments don't support such artifact-centered discourse well. In this paper we report on the design of Kukakuka, a system that addresses this problem by associating discussion groups with online artifacts in the form of a web pages. Our work with Kukakuka illustrates issues that arise in design for artifact-centered discourse, particularly when the implementation is constrained to standard Internet technologies.

Keywords: web-based collaboration, discussion tools, document annotation

1 Introduction

There is a significant trend toward computer mediated distance collaboration in educational contexts ranging from K-12 through university to the life-long professional development of adults. This situation presents a great responsibility on the part of software developers to provide technologies that are specifically designed to support effective learning activities. Most human discourse in learning and work references relies on external artifacts as resources or aids for conversation, but many online learning environments don't provide adequate representational and computational aids in support of such discussions. In this paper, we report on the design of a system that addresses this problem by linking discussion groups with online artifacts in the form of web pages. The system is distinguished from similar efforts by how it supports conversations about collections of related web pages. In the remainder of this introduction, we motivate our system's design with a discussion of the relationship between discourse tools and the artifacts that are the topics of discussion.

1.1 Artifact Centered Discourse

Two important contributors to effective learning activities are social processes such as collaboration and mentoring, and the use of representational aids. We focus specifically on the combination of these contributors. In face-to-face learning, participants often reference and manipulate external artifacts (sketches, pictures, and objects). Interaction with the artifacts is an essential part of the activity. This should also be true of online learning. For example, students enrolled in online courses in computer science will need to share and discuss the artifacts of software engineering (e.g., interface mockups, class diagrams, source code) as they collaborate with

faculty and with other students on the design, implementation, and evaluation of innovative software technology. Educators seeking curricular materials by which they can address local content and performance standards will need more than a searchable database of "learning objects." They will also want to discuss the use of these objects with each other and their authors. Educators engaged in professional development may need to examine examples of student work as they learn to recognize indicators of student learning, or to observe and discuss videos of exemplary instruction.

Yet the online world provides only impoverished support for discourse about artifacts. Many commercial products for online learning are not designed to establish and carry on a discussion in the context of a visual artifact that can be annotated and referenced. Some tools support annotation of textual artifacts, but do not support extended discourse beyond localized comments. Other tools, quite similar to our own work, enable threaded discussions to be associated with web pages. Threaded discussions present their own problems apart from the issue of coordination with artifacts. They are notorious for their lack of convergence, a situation that has been blamed on the fact that a tree constructed from reply relations is an inherently divergent representation (Hewitt, 1997). Furthermore, the representation is based on the historical development of the discussion rather than its conceptual content, making it difficult to quickly grasp and assess the status of the discourse and hence to make contributions that move it forward (Turoff, et al., 1999). Coherence problems arise when multiple threads of discussion intertwine within a single discussion space (Herring, 1999).

Our laboratory is engaged in several lines of work intended to address these and related problems (Suthers, 2001; Takeda & Suthers, 2002). In this paper, we focus on the problem of designing the relationship between online discussions and the artifacts under discussion. We use the phrase artifact-centered discourse (Suthers, 2001) to indicate the type of discussion and argumentation that should be supported. These discussions are also called *Anchored Discourse* (Guzdial, 1997), *Contextualized Communication* (TECFA, 2001), or *Document-Centred Discourse* (Sumner & Shum, 1998).

Consideration of learning applications such as those mentioned above lead to the following desiderata. Online discussion environments should allow the inclusion of visual artifacts, such as data graphs, videos of teacher and student work, or problem solutions under construction (e.g., knowledge maps, software designs, etc.). These artifacts should exist outside of individual messages (i.e., attachments are not sufficient), and stay visible during the conversation. Participants should be able to change the artifacts under discussion in a natural way, leading to a new discussion space if necessary. Ideally, it will be possible to carry on a discussion about a collection of artifacts. Participants would be able to browse through the artifacts, and refer to individual artifacts (ideally, parts of artifacts) in their contributions. Each discussion space would be focused on a single artifact or collection of artifacts, addressing some of the coherence and divergence problems of threaded discussions. We now take a closer look at several major design approaches.

1.2 Relationships between Discourse Representations and Artifacts

In typical tools for online learning (such as Blackboard and WebCT), discussion tools and shared artifacts are displayed on entirely different screens. This is hardly conducive to online discourse about artifacts. One can work around this problem by manually opening two windows and placing the discussion tools next to the artifacts under discussion. We call this arrangement *parallel artifact-centered discourse* or *parallel ACD*. There

is no communication or coordination between the discourse and the artifacts; they are simply displayed in parallel. The burden of setting up the display can be taken off the user by binding the artifact and the discourse tool together in one display. A good example of this approach is Ubiquitous-D3E (Sumner & Shum, 1998). A D3E workspace displays a web page in one frame and an HTML interface to a threaded discussion in another frame. Our system takes a very similar approach, but with one important difference (to be discussed below) concerning the relationship between discussion groups and web pages.

An *embedded* discourse representation embeds comments directly in the artifact under discussion, or achieves a similar effect by inserting comments for display without modifying the original artifact. Examples include the Annotation Engine (Berkman, 2001), CoNote (Davis & Huttenlocher, 1995) and CritLink (Yee, 1998), all of which take a proxy server approach. (An entirely different approach is a Wiki Wiki, which enables full user editing of documents via the web. See <http://wiki.org/> and <http://www.c2.com/cgi/wiki?WikiWikiWeb>.) Because the discourse always takes place in the context of the artifact, *embedded ACD* has the advantages that it is easier to refer to parts of the artifact or to recover the portion of the discussion that is concerned with a given part. Some empirical evidence supports this approach. Guzdial (1997) compared "anchored collaboration" (embedding links to discussion threads within artifacts of interest) to newsgroups in undergraduate classes, finding that longer threads of discussion took place with anchored collaboration. Wojahn, et al. (1998) performed an experimental comparison of "split-screen," "interlinear" (embedded) and "aligned" (side by side) interfaces for collaborative annotations, and found no significant difference in time to task completion but significantly fewer communications in the split-screen representation, which apparently presents the greatest distance between the artifact and the annotations and hence cognitive load in processing them. The fact that the discourse is distributed across the artifact in embedded ACD leads to one of its disadvantages: it is more difficult to get a sense of the whole discussion or to notice relevant relations between discussions about different parts of the artifact. The artifact may also become cluttered with comments.

The tradeoff between parallel and embedded ACD can be resolved by providing discourse and artifacts with independent representations optimized for the needs of each, but providing a logical linkage between the two. In *linked ACD*, discourse representations and contextual artifacts are displayed side by side, as in parallel ACD, retaining the reply structure and chronology of the discourse. The difference is that all contributions are referenced to the appropriate artifact or part thereof. Selection of a discourse contribution would cause the object discussed by that contribution to be displayed or highlighted, disambiguating terms such as "this" and "that." Selection of an artifact may also bring up or highlight the discourse contributions concerned with that artifact. This approach resolves the tradeoff between the unrelated representations of parallel ACD tools and contextualized yet conversationally fragmented discourse contributions of embedded ACD tools. Examples of linked ACD systems include the Journal of Interactive Media in Education (Sumner & Shum, 2001), Pink (Takeda & Suthers, 2002) and Kukakuka, the system described in this paper.

We believe that linked ACD, properly designed, can improve coherence and convergence of artifact-centered discourse by collecting together topically related contributions. This is a general solution, applicable to any online artifact-centered discourse, but we are particularly interested in its applications to supporting knowledge-building discourse within learning communities. The crucial issues in the design of linked ACD pertain to the internal structure of a discussion workspace: at what granularity does the linkage take place, and how are distinct yet related artifacts and their discussions organized with respect to each other? Our experimentation with our own

linked ACD system has provided some insight into these issues.

1.3 Kukakuka

Theoretical investigations should be complemented with explorations in practice. We have been pursuing a strategy of rapid, iterative prototyping of a system for artifact-centered discourse and early and continual use of this system to support the first author's professional teaching activities. The remainder of this paper describes the design of *Kukakuka*, a web-based interface for threaded discussion of web pages. ("Kukakuka" is Hawaiian for "discussion." It is properly spelled with macrons over the "u" and "a" to indicate long pronunciation.) Our work with Kukakuka illustrates issues that arise in design for artifact-centered discourse, particularly concerning the relationship between messages and artifacts. We also explore implementation issues that arise when the implementation is constrained to standard Internet technologies.

Since the first prototype became operational, Kukakuka has been used in support of an upper-division course on Human Computer Interaction. Artifacts under discussion include sample web sites and screen dumps of example interfaces, students' short essays, solutions to small design assignments, and more extensive documentation of their semester-long projects, as well as the Kukakuka system itself. Some aspects of the design of Kukakuka itself were developed in class, using both formal representations such as dataflow and entity-relationship diagrams, and informal representations such as quick sketches of usage scenarios that result from alternative designs. Discussions with students and problems they encountered in use of the system contributed significantly to Kukakuka's design.

Of particular relevance to the foregoing discussion was a design flaw that we identified almost immediately after beginning to use the system. Like Ubiquitous D3E, Kukakuka initially associated an entire discussion group, including multiple threads, with a single web page. The assumption was that each discussion would be about a single artifact. Yet we soon discovered what in retrospect is obvious: web artifacts are structured entities, consisting of multiple parts (pages), and the linkage between a discussion and an artifact should take place at a granularity that references the parts being discussed. We discovered this because the first discussion we set up was for an entry page containing links to the individual pages that students had constructed in response to an assignment. All of these individual pages properly belonged to the same discussion group, as the topic was the same and one might want to make comments that transcended the individual responses. However, much of the discussion needed to reference particular students' contributions; i.e., pages linked to by the entry page. Users had to identify the relevant page in their subject lines, e.g., "Kent's summary," so that other users could manually navigate to the referenced page. We immediately realized that individual messages should be linked to the component pages that they reference (or to the group page if the message concerns the collection of pages as a whole).

The decision to allow the messages in a discussion reference multiple artifacts led to the question of whether there should be some kind of structure to this linkage, or whether any message could reference any page. We tried a prototype that allowed any message to specify its own artifact. We also compared alternate scenarios by sketching them on a whiteboard equipped with a Mimeo™ recording device, and then reviewing them using the Mimeo™ replay mechanism. Through these activities and consideration of the problems of divergence in threaded discourse, we arrived at the conclusion that each thread should be associated with a single artifact.

Messages within a thread are not allowed to change the artifact. This design decision is somewhat controversial. When asked, users tell us that they want to be able to change the artifact associated with their messages. However, we refrained from providing this functionality because to do so might introduce the problems of topic drift that we are trying to avoid. If a user could change the artifact in mid-thread, then thread topics could "drift" to artifacts that were not the original topic, potentially leading to unfocused discussions. Even if users were disciplined in avoiding this problem, another problem would be unavoidable: other users could not tell by looking at the start of a thread which artifacts were discussed in that thread. They would have to search the entire thread for possible references to a given artifact of interest. This problem could be solved by providing an index of messages by artifact, but to avoid topic drift we elected to restrict the default context of a thread to a single artifact. In the future we can add indexing of messages by the artifact that they reference within the body of the message using conventional hypertext references.

In the remainder of this paper we discuss the high level design of Kukakuka; provide a use scenario; outline the implementation level design; describe its use in a computer science class; and discuss issues and implications.

2 Design

Kukakuka was designed to meet the following criteria:

1. ***Linked ACD:*** Threaded discussions are associated with artifacts while maintaining the independent display of each.
2. ***Focused Discussions:*** The system should discourage topic drift. Once an artifact has been identified as the topic of discussion, the discussion should stay focused on that artifact. (But see next item.)
3. ***Multiplicity of Artifacts:*** Workspaces should support the discussion of multiple related artifacts in an organized manner (e.g., without loss of focus).
4. ***Guarantee of Context:*** The artifact under discussion should always be made visible when reading or contributing to a discussion. Corollary: The artifact and its discussion should be displayed in a single window to minimize problems for inexperienced users. Multiple window versions may be provided for expert users.
5. ***Universal Artifacts:*** The system should support discussion of any web page on the Internet regardless of location and without needing to copy it to the home server.
6. ***Ease of Access on Multiple Platforms:*** The system should be accessible to any standard web browser without requirements such as client side Java or plug-ins that would discourage use by technically unsophisticated and overworked users (e.g., public school teachers).
7. ***Interoperability and Reuse:*** To the extent possible, the system should build upon existing newsgroup server technology without modification and without impairing the ability of other types of clients to access the newsgroups.

2.1 Conceptual Architecture

The conceptual architecture of Kukakuka's current implementation is summarized in Figure 1. A user connects to a single discussion group at a time. (In the near future, users will authenticate as members of user groups, and discussion groups will have access rights defined for each user group.) Each discussion group has a contextual or "home" artifact associated with it that provides the global context for the group discussion. (An "artifact" is a reference to some resource on the WWW.) Each group potentially contains many discussion threads. Each thread

also has a contextual artifact, which defaults to the group contextual artifact unless specified otherwise. Each thread contains potentially many messages. Importantly, the context of every message in a thread is the home artifact of the thread. As previously discussed, this constraint helps prevent topic drift and makes it easier to identify all portions of the discourse that reference a particular artifact. However, we recognize that users will want to be able to reference other artifacts in their discussions, e.g. for comparative purposes. Hence the conceptual architecture allows users to reference other artifacts by embedding these references in the body of their messages. Currently the software does not directly support this operation, although a technically savvy user can accomplish the desired effect by embedding HTML links in his or her message. We plan to provide support for insertion of such links in the future.

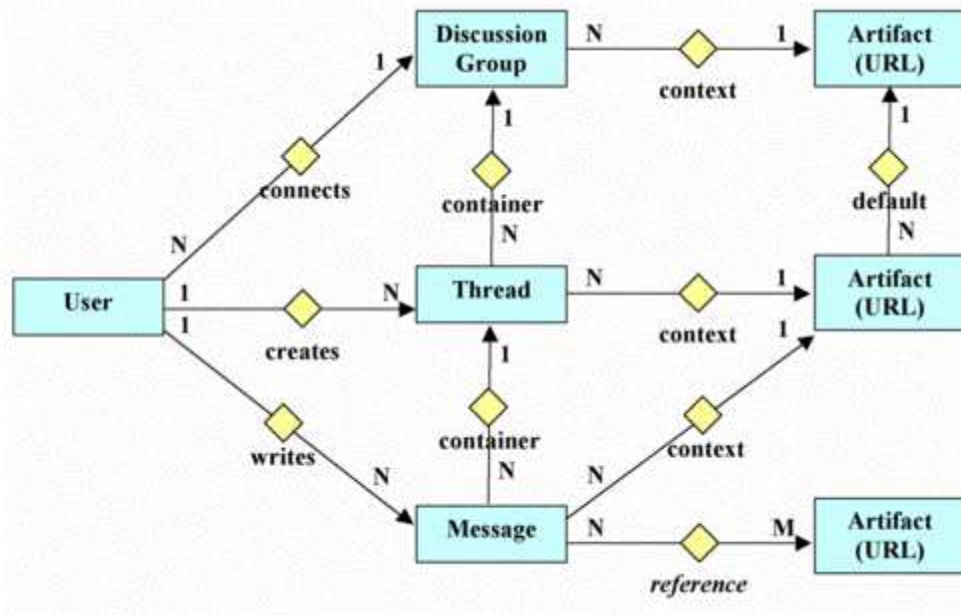


Figure 1. Conceptual Architecture

2.2 Abstract Implementation Architecture

The implementation architecture is shown at its coarsest granularity in Figure 2. (The implementation will be detailed later in Figure 7.) Generally speaking, the implementation consists of a collection of servlets that bind together newsgroups of an external NNTP server (Kantor & Lapsley 1986) with web pages of an external WWW server, and present the combined contents of these information sources in a single web page accessible to a browser. A database will provide the servlets with authentication capabilities and user history information (e.g., unread messages). Database functionality is being implemented at this writing.

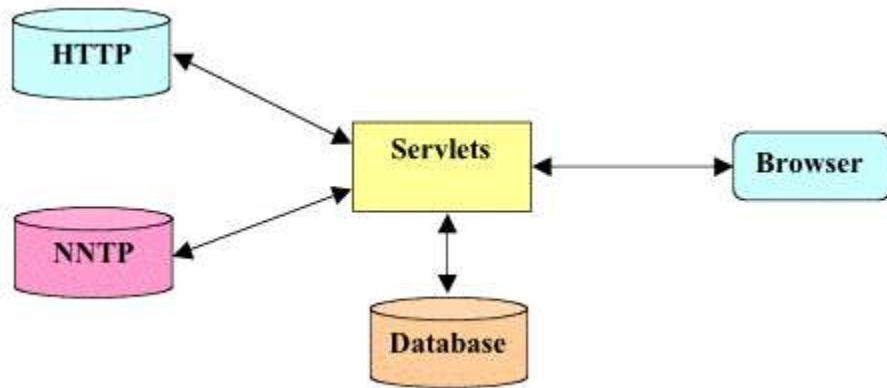


Figure 2. Abstract Implementation Architecture

3 Sample User Scenario

3.1 Configuring a Kukakuka Server

To configure a Kukakuka server, an instructor/administrator creates a configuration file containing the location of the NNTP server to be used and the path to the discussion group that will serve as the root for this server. The configuration file can also contain HTML for identifying and welcoming information, news, etc. A Kukakuka server may be provided with more than one configuration file, thereby supporting multiple communities. The Kukakuka server uses the information in a configuration file to generate the entry page. Figure 3 shows a sample entry page. The provided HTML is displayed on the left side, and all discussion groups falling under the designated root group are listed on the right side on the entry page. A user can click on one of the group links to enter the group page.

Welcome to the Kukakuka server for ICS463

Intro to Human Computer Interaction Design

Instructor: Dan Suthers, suthers@hawaii.edu
Kukakuka developer: Jun Xu, junx@hawaii.edu

News

A new version was installed 11/16/01 with some minor improvements in navigation tools. We are working on a more fundamental redesign of the internals.

Usage Notes

Use HTML for formatting. Otherwise it all runs together.

Use your email address for From and giving an informative Subject.

To create a new thread, select NEW_THREAD and manually paste the URL of the page to be discussed into the Reference field. Once you have done this anyone replying to your message will have their message automatically associated with the URL you specified.

The "back" and "forward" buttons in the artifact frame only work in Netscape 4.7.

Please send feedback to kuka-developers@hawaii.edu if you have suggestions.

➤**ics463**

➤**2001-fall**

[assignment-2](#)

[assignment-3](#)

[assignment-4](#)

[assignment-5](#)

[assignment-7](#)

[gallery](#)

[nielsen-alertbox](#)

[projects](#)

- [auto-fluid-monitor](#)
- [bc-chat-system](#)
- [bulletin-board](#)
- [clickage-dot-org](#)
- [course-scoop](#)
- [hawaiian-jewelrey-store](#)
- [ics-map-system](#)
- [instant-messaging](#)
- [japanese-tutor](#)
- [online-purchasing](#)
- [quizbank](#)
- [tours-ru](#)
- [trouble-call-management](#)

➤**assignment-1**

➤**kuka**

➤**test**

Figure 3. Kukakuka Entry Page

3.2 Creating a Discussion

Creation of a new discussion workspace within an existing Kukakuka installation requires the following actions: (1) Create the NNTP discussion group within the designated hierarchy; (2) post a message to this NNTP group with a specific subject line (defined by the Kukakuka installation) and containing the URL of the group's home artifact in the body. Henceforth this artifact will be retrieved and displayed whenever one enters the group.

[Back](#)
[Forward](#)

[KUKAKUKA_HOME](#)
[GROUP_HOME](#)
[NEW_THREAD](#)
[RELOAD](#)

ICS 463: Intro to Human-Computer Interaction Design, Fall 2001

General Info

Assignment 1 (Due Sept. 19th)

[Overview](#)

[Readings](#)

[Assessment](#)

[Instructor](#)

[Resources](#)

Course Management

[News](#)

[Schedule](#)

[Assignments](#)

[Projects](#)

Name	Making The Case
XXXXXXXXXXXX	html 9/19
XXXXXXXXXXXX	html 9/19
XXXXXXXXXXXX	html 9/19
XXXXXXXXXXXX	html 9/19

Subject	From	Date
Discussion of Assignment 1	Dan Suthers	27 Sep 2001 10:06:47 GMT
UJD comments	XXXXXXXXXX	27 Sep 2001 16:54:15 GMT
XXXX's exec summary	XXXXXXXXXX	27 Sep 2001 21:02:11 GMT
XXX's summary	XXXXXXXXXX	1 Oct 2001 11:40:48 GMT
General Review	XXXXXXXXXX	1 Oct 2001 20:37:57 GMT
XXXXXXXXXX Exec Summary	XXXXXXXXXX	2 Oct 2001 05:30:25 GMT

----no message----

Figure 4. Kukakuka Group Page

3.3 Reading a Group's Messages

Each group's page shows the group artifact (a web page) in the large left frame and a tree view of all the messages in this group in the right-middle frame. Figure 4 shows the group page that originally motivated the redesign to link each thread to a component artifact. The user can click any message and read the content of this message in the right-bottom frame. At the same time, the home artifact of the message's thread will be shown in the left frame, as shown in Figure 5.

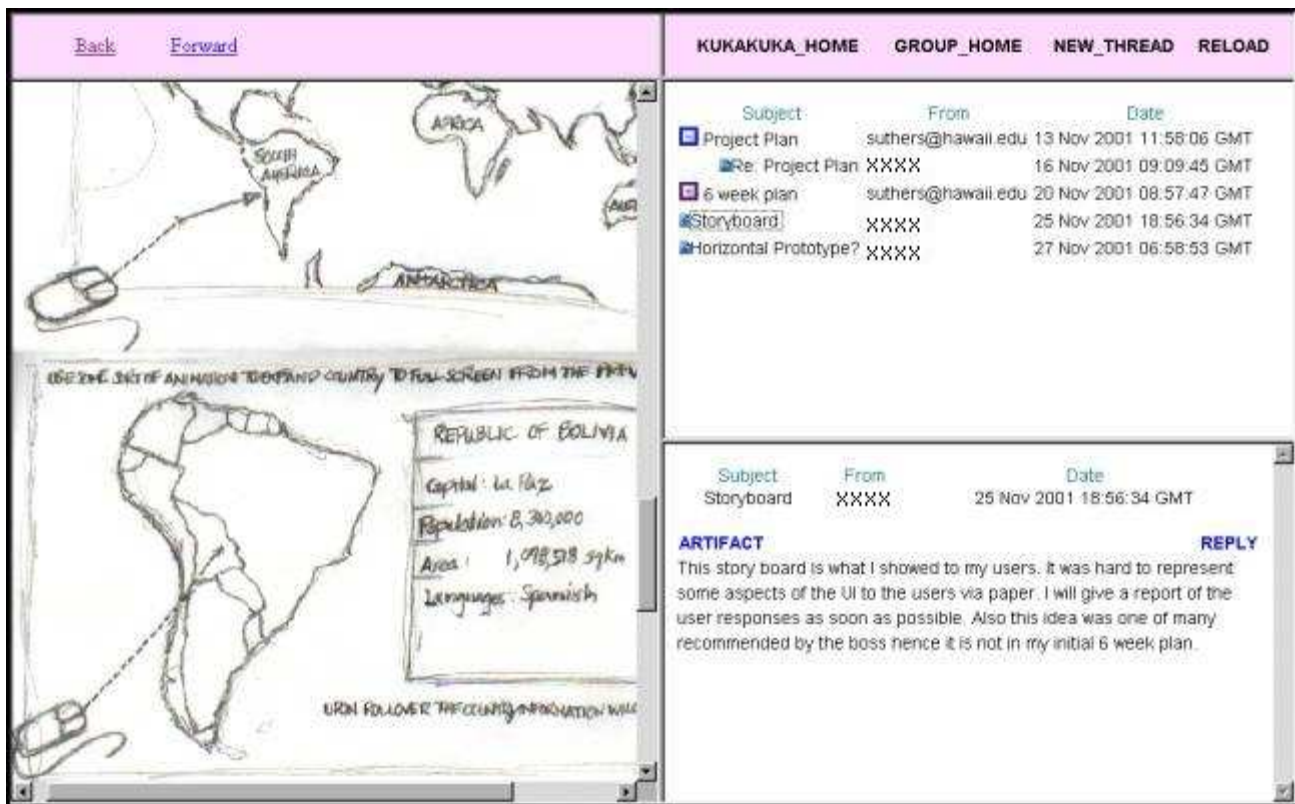


Figure 5. Viewing a Thread's Contextual Artifact while Reading a Message

3.4 Creating a Thread

A user creates a new thread and (optionally) associates a new home artifact with it as follows. The "NEW_THREAD" link in the menu frame causes a new thread entry form to show up in the right-bottom frame, shown in Figure 6. The Reference field holds the URL for the group contextual artifact as a default. The user can optionally modify this URL or paste in a new one. (We would like to enable users to set this URL to an artifact displayed in the left frame at the click of a button; however, security restrictions in browsers disallow us from obtaining information about the contents of other frames.) After entering "subject", "email address" and the message contents, the user can select the "Send it now!" (submit) button and the new message will be posted. Henceforth, all replies to this message and other messages within the new thread will have the indicated contextual artifact. Therefore, when users reply, they do not need to input a reference.

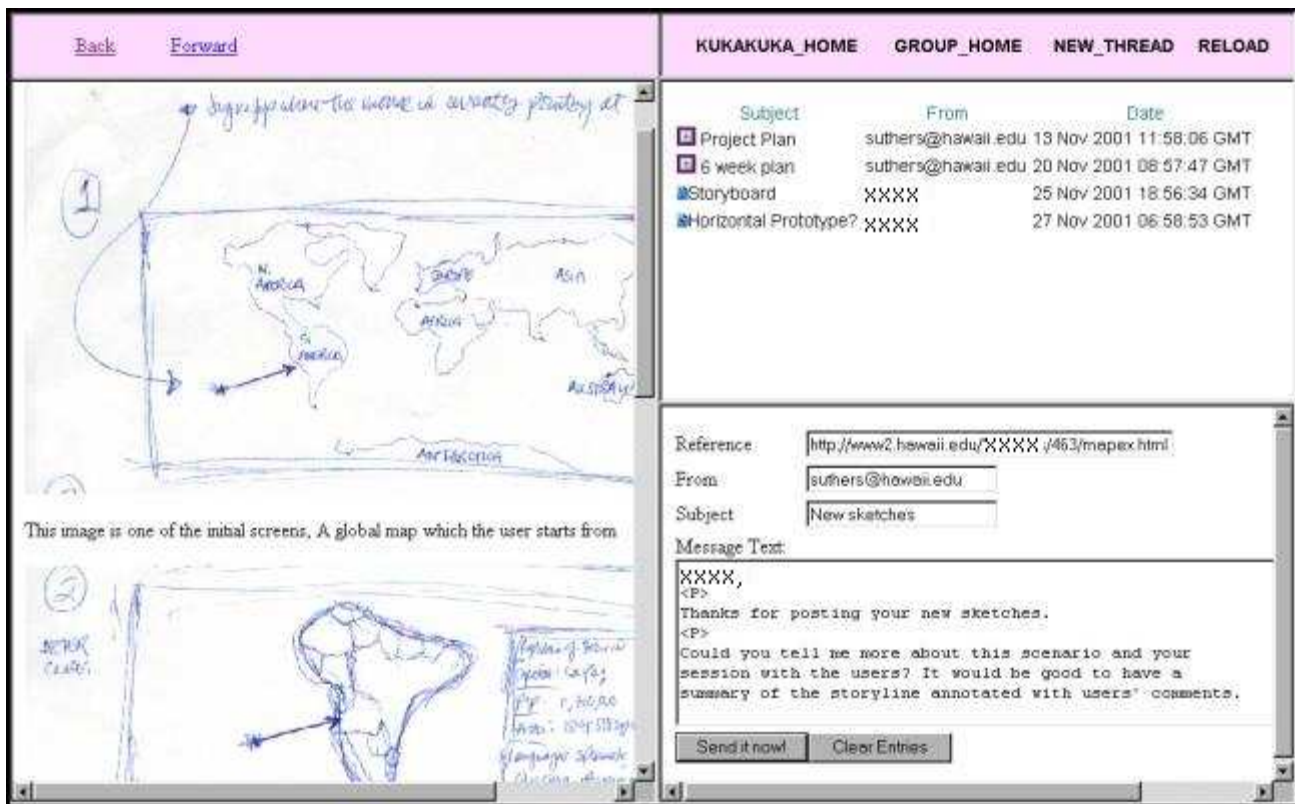


Figure 6. Starting a New Thread with a Contextual Artifact

Other navigational tools are also provided. The user can go back to the system entry page by clicking "KUKAKUKA_HOME" link and go back the group home page by clicking "GROUP_HOME" link. "RELOAD" will reload the message threads. "Back" and "Forward" change the contents of the artifact frame as indicated without modifying the contents of the other frames. (If a user alternated between browsing and working on a message, the browser's Back and Forward buttons could remove a message that was in progress rather than change the artifact frame, as was probably intended.) The current implementation of frame-specific Back and Forward relies on a frame-specific history, which is not present in some current browsers.

4 Detailed Implementation Architecture

As shown in Figure 7, the currently implemented system uses Java Servlet technology. The system is divided into several servlet modules, each of which takes responsibility for one kind of functionality. For instance, the msglist servlet generates a message tree for one group. All browser actions initiated by the user are accomplished via servlet doGet() /doPost() methods, then servlets connect to the NNTP server or database, and generate different interfaces on the web browser.

The News servlet generates a template for making new Kukakuka home pages, which enables instructors to add their own HTML for customizing the appearance of the page, and to specify the news server and group as the entry point (root). The server/group name and HTML source are stored in a configuration file. The servlet will: (1) read the configuration file and get server information, (2) connect to NNTP server and generate the subgroup tree.

GroupPage servlet generates the group entry page, which shows the group artifact on the left side, and the message list and message body on the right side. The URL of the group artifact is stored in one message with a special subject line. The servlet will try to find this message and read its body.

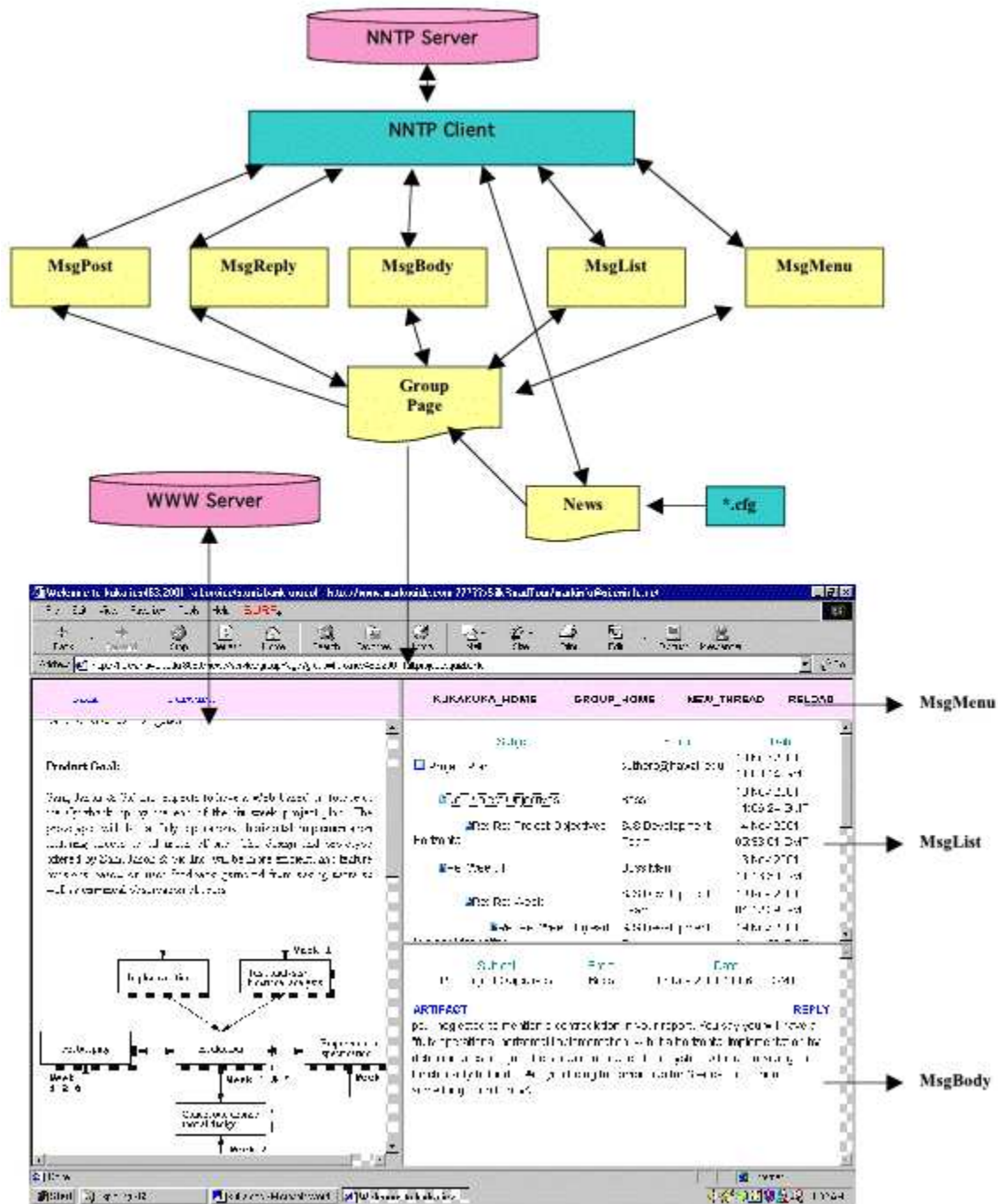


Figure 7. Detailed Implementation Architecture

The Msglist servlet shows all the messages in one newsgroup in a tree structure. A "+" image will be shown in the front of one message if it has replies. If the "+" is selected, the thread is expanded and a "-" image is shown. The servlet caches all the messages in a tree structure, then displays them according to different requests from the

browser, such as expanding one message with all the replies. The "RELOAD" link forces the servlet to reload this tree structure.

In order to associate contextual artifacts with threads (and, potentially, individual messages) without impairing the ability of other types of NNTP clients to access the newsgroup, we elected to define a new message header called "Artifact." When a user posts a new thread, the URL of artifact will be stored in this header. When readers reply this message, the same artifact will be the default reference of the reply. The Msgbody servlet checks the message header and sees whether this header exists. If so, Msgbody shows the artifact in the left frame.

Our method of using a new header not only prevents interference with other clients of the newsgroup (if they are allowed to connect: see discussion of security below); it also permits other clients to utilize the artifact reference. For example, if one connects to our NNTP server with the Netscape newsreader configured to "show all headers," one can see the Artifact header and click on its contents to cause the contextual artifact to be loaded in the Netscape web browser. We believe that an Artifact header is a simple yet powerful way to contextualize messages that can be supported by a variety of news and mail tools. None of the existing NNTP headers allow URLs as their contents. Therefore we are considering proposing this header as an extension to RFC 1036 (<http://www.w3.org/Protocols/rfc1036/rfc1036.html>).

5 Discussion

Kukakuka was used in the fall of 2001 to support artifact-centered discussions in an undergraduate Human-Computer Interaction class. Kukakuka underwent continual revision during this time, and was itself a topic of class design discussions.

5.1 Evaluation of Classroom Use

Excluding test messages and discussions of Kukakuka itself, 458 messages were posted in 22 groups devoted to the students' work. Although we have not undertaken formal analysis of newsgroup contents, an informal examination yields some fundamental conclusions. Messages are overwhelmingly "on task," addressing the artifact at hand with little topic drift. Threads tend to be short, quickly exploring the most salient conclusions about an artifact before participants move on to the next version of the design as expressed by a subsequent artifact and its associated thread. Thus the focus of the discussion is maintained as much by the ability to form a new thread centered on a new artifact as it is by the restriction that all messages within a thread are displayed along with the single artifact for that thread. The ability to handle multiple related artifacts within a single discussion space is critical, a significant advantage over the current design of Ubiquitous D3E (Sumner & Shum, 1998), but an advantage shared by JIME (Sumner & Shum, 2001) and Pink (Takeda & Suthers, 2002). (JIME is oriented towards documents on the server rather than external web pages; Pink handles both.) Overall, the system is proving very useful in supporting higher levels of communication between the instructor and class members concerning student design projects as captured by the artifacts that students post on the Web.

Problematic aspects of our experience are attributable to several sources: usability issues in the interface itself; the lack of user-specific persistent state; and the need to better integrate this tool with existing work practices of our students. These issues are discussed in the next section.

5.2 Implementation Issues and Future Work

Some of the usability issues have been or will be resolved by revisions that are feasible within the current implementation strategy. For example, the opening and closing of threads of message headers can be more smoothly handled on the client side (e.g., using DHTML), rather than by the servlet, which forces a reload of the frame and hence loss of the current scroll position. Clarifications and reorganization of the command links are also in progress. Yet other issues remain that will require more powerful technologies to overcome the limitations of web browsers. We are particularly hampered by the security restriction that client-side processes such as Javascript cannot identify the URL that is currently being displayed within a frame, and by the lack of frame-specific histories in some browsers. These limitations prevent us from (1) providing facilities for navigating within the artifact frame without affecting other frames; (2) checking whether the contextual artifact is already displayed before forcing a reload of it when opening a new message; and (3) providing single-click solutions (as opposed to manual pasting of URLs) to associating a new thread with the currently displayed artifact or inserting references into the body of a message. An applet-based interface would solve these problems. Although we may pursue such a solution, we will maintain a line of development that avoids client-side Java because of the wide variation in versions of Java supported by browsers and the difficulties that some members of one of our targeted user populations (public school teachers) may experience in dealing with browser configuration issues.

Apart from the reloading of the thread listing, the most frequent complaint by users is the lack of assistance in separating previously read from new messages. With the addition of a database and user-specific authentication, currently underway, we will be able to keep track of read/unread messages for each user and distinguish these in the display. User authentication will also help restrict visibility of groups to those relevant to the user.

The third area for improvement, connection to users' normal work practices, will be addressed by the database as well. Our users use email frequently, yet may forget to check the Kukukuka web-based environment for new messages. The improved server will be configurable to mail out periodic (such as weekly or daily) listings of newly posted messages, including the subject headers and a URL that takes the user directly from their mail system to the message in question. This facility will require that we address the problem of providing each message with its own URL without forcing the entire display to reload every time a new message is read (currently only the message frame and artifact frames are reloaded).

Previously we mentioned that one advantage of our design is that it does not interfere with other methods of accessing the same NNTP groups. Users can view them using standard newsgroup readers, and even access the artifacts by displaying all headers to expose the Artifact link. In fact, Kukakuka servers can be set up to access external NNTP servers not owned by the Kukakuka administrators. However, for security reasons, administrators of combined Kukakuka/NNTP servers may elect to restrict access to the NNTP server to only the servlets. Security will be handled in two layers: (1) Authentication to the Kukakuka servlets, checked against the database. (2) Web page/Server/Client authentication, using SSL. Similarly, access to the database may be restricted to the Kukakuka servlets. In our implementation, a connection pool mediates this access. We have conducted empirical testing showing that a connection pool speeds up database access.

5.3 Conclusions

Much of human discourse is centered on representational artifacts, yet web based applications for learning don't support such discussion well. Given that social interactions and representational aids both play a crucial role in learning, this is a problem worth addressing. The solution is simple yet potentially powerful. Discussion groups and discussion threads are associated with web-based artifacts. Each message's artifact is always visible when reading and posting messages. The implementation uses a straightforward architecture that can attach to any external HTTP and NNTP servers without modification beyond the requirement that the NNTP server accept the additional Artifact header where we store the URL. Although the implementation architecture permits total flexibility in associating each message with its own artifact, we have chosen to require that all messages in a thread share a single artifact. This constraint guarantees that the context of the discussion is available without any special effort, and helps keep the discussion focused on the topic at hand.

Kukakuka is designed with the objectives of being easy to use, applicable to any web page, and running within any web browser without plug-ins or other special configurations. These objectives constrain our design somewhat yet make the system usable by a broader community. The system has proven useful in supporting asynchronous discussions of projects and examples in a course on human-computer interaction. Many of the usability problems we identified are being addressed through client side management of certain dynamic interface functions such as the thread listing, and though the personalization that will be possible with the addition of a database and authentication. Unresolved usability problems stem largely from security-motivated limitations on the ability of processes in the browser to inspect browser contents. The fundamental tradeoff is between improved interface functionality versus the ability to function within de-facto standard browsers without special configuration.

We believe that the most promising area for further research in the design of software for artifact-centered discourse lies in the coordination of multiple representations, including discourse representations and disciplinary artifacts, in accordance with the flow of learning activities. As we continue to use the software for different learning applications we will study ways in which users need to refer to, group, and move between web artifacts during the course of an extended discussion or investigation.

6 Acknowledgements

The first author was supported by National Science Foundation grants 9873516 and 0093505. Opinions expressed are those of the authors and do not necessarily reflect policy of NSF. We are grateful to the support of many of our colleagues in the design and implementation of Kukakuka, including Bruce Harris, Dan Jones, Joshua Kuo, Trudy Lin, Bin Ma, and the Fall 2001 class members of ICS 463 Human Computer Interaction.

7 References

Berkman Center for Internet and Society at Harvard Law School (2001).

Annotation Engine. Online: <http://cyber.law.harvard.edu/projects/annotate.html>.

Davis, J.R. and Huttenlocher, D.P. (1995).

Shared Annotation for Cooperative Learning. *Proceedings of Computer Support for Cooperative Learning Conference (CSCL '95)*, pp.84-88. Online: <http://www-cscl95.indiana.edu/cscl95/davis.html>.

Guzdial, M. (1997, December).

Information ecology of collaborations in educational settings: Influence of tool. *Proceedings of the 2nd International Conference on Computer Supported Collaborative Learning (CSCL '97)*, pp. 83-90. Toronto: University of Toronto.

Herring, S.C. (1999, January).

Interactive coherence in CMC. *Proceedings of the 32nd Hawai'i International Conference on the System Sciences (HICSS 32)*. (CD-ROM). Maui, Hawai'i: Institute of Electrical and Electronics Engineers, Inc. (IEEE).

Hewitt, J. (1997).

Beyond threaded discourse. Paper presented at *WebNet'97*. Online: <http://csile.oise.utoronto.ca/abstracts/ThreadedDiscourse.html>

Kantor, B. and Lapsley, P. (1986).

Network News Transport Protocol, RFC 977, Online: <http://www.w3.org/Protocols/rfc977/rfc977.html>. (Common extensions are described in RFS 2980, <http://jakarta.apache.org/james/rfclist/rfc2980.txt>.)

Sumner, T. and Shum, S. B. (1998).

From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. *Proc. CHI'98: Human Factors in Computing Systems* (Los Angeles, 18-23 April, 1998). ACM Press: New York. Preprint: <http://kmi.open.ac.uk/tr/papers/kmi-tr-50.pdf>. See also <http://d3e.sourceforge.net/ubiquitous-d3e.html>

Sumner, T. and Shum, S.B. (2001).

JIME: An Interactive Journal for Interactive Media. *First Monday*, 6, (2), Feb. 2001. Online: http://firstmonday.org/issues/issue6_2/buckingham_shum/

Suthers, D. (2001).

Collaborative Representations: Supporting Face-to-Face and Online Knowledge-building Discourse. *Proceedings of the 34th Hawai'i International Conference on the System Sciences (HICSS-34)*, January 3-6, 2001, Maui, Hawai'i (CD-ROM), Institute of Electrical and Electronics Engineers, Inc. (IEEE). 2001. Preprint: <http://lilt.ics.hawaii.edu/lilt/papers/2001/Suthers-HICSS-2001.pdf>

Takeda, T. and Suthers, D. (2002).

Online Workspaces for Annotation and Discussion of Documents. Poster presentation, *WWW 2002*.

TECFA (2001).

TECFA Newsletter, October 2001, School of Psychology and Education, University of Geneva. http://tecfa.unige.ch/tecfa/newsletter/TecfaNewsletter_oct.pdf

Turoff, M., Hiltz, S. R., Bieber, M., Fjermestad, J., & Rana, A. (1999).

Collaborative discourse structures in computer mediated group communications. *Journal of Computer Mediated Communication*, 4(4). Online: <http://jcmc.huji.ac.il/>

Wojahn, P.G., Neuwirth, C.M., & Bullock, B. (1998, April).

Effects of interfaces for annotation on communication in a collaborative task. *Proceedings of the Conference on Human Factors in Computing Systems (CHI '98)*, (pp. 456-463). Los Angeles: ACM Press.

Yee, K., (1998).

CritLink: Better Hyperlinks for the WWW. *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, June 20-24, 1998, Pittsburgh, PA, USA. ACM 1998 1998. Preprint: <http://www.crit.org/~ping/ht98.html>